ISSN 0956-9979 JUNE 1997

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

Technical Editor: **Jakub Kaminski**

Assistant Editor: **Megan Skinner**

Consulting Editors:

Richard Ford, IBM, USA
Edward Wilding, Network Security, UK

IN THIS ISSUE:

- **Administrative problems.** This month, *VB* takes a look at the issues of virus protection and recovery – not, as previously, from the end-user's point of view, but from that of the administrator. See p.13 for Phil Crewe's analysis.
- **A multi-edged sword.** A new series of multi-platform reviews debuts in this issue, and *McAfee's* offering is first on the scene. Turn to p.21 for the low-down.
- **Headline breakers.** Both this month's news stories concern legal issues, and both involve *McAfee*: what has the company been up to now? All is revealed on p.3.

CONTENTS

GUEST EDITORIAL

NEWS

INSIGHT

VIRUS ANALYSES

FEATURE

PRODUCT REVIEWS

VIRUS BULLETIN **©1997 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England.**
Tel +44 1235 555139. /97/$0.00+2.50

GUEST EDITORIAL

The Update on Updates

'Our product will protect against all known and unknown viruses. Thanks to our unique technology, you too can enter an eternal state of security nirvana with no need to update your anti-virus protection'. Does this ring a bell? This kind of over-enthusiastic marketing-speak, often seen in the last few years, would rarely engender genuine concern for confused users. Now, an ironic smile is enough to 'comment' such statements. Flames are thrown only when someone tries to back the advertising stunts with 'technical proof'.

Until there is a radical change in the most commonly-used operating systems, the anti-virus industry cannot stand still. Anti-virus technology remains bound to its estranged partner, virus-writing technology – almost a classic love/hate relationship. It can stay in step with, or even anticipate, the enemy's moves, but as long as new virus-writing methods are developed, and new opportunities arise, anti-virus products must constantly be revised to fit into an ever-changing reality.

“ giving users access to daily updates is becoming a necessity for those who want to stay in business ”

What if the anti-virus industry becomes obsolete? – a scary thought indeed. And if all virus writers find other ways of leaving their mark in life? Changes and upgrades to operating systems would probably be enough to induce changes in anti-virus products, but as long as virus writing is 'cool' or a challenging thing about which to boast, the industry has busy days ahead.

Can we re-educate virus writers, and convince operating system producers to stop developing new versions? (For this to happen, developers must first believe users are happy with what they have been given…) Until then, updating is a must! If you don't like updates, you have to put up with upgrades. Because your dreams about a once-installed, everlasting protection are shattered by reality, you accept the necessity of updating your anti-virus programs.

If you want to control the way the product you use is updated, and the time this takes, then you want all updates to be available at any time you decide to install them. These days, easy access means one thing: the Internet. Web pages or FTP sites provide the most common electronic method of obtaining the latest versions of software products and the latest information. If you think you have more important things to do than organizing regular updates, you may want to rely on your supplier to send them to you. Traditionally recognized as a standard and secure way of distributing updates, sending diskettes through the post has served customers well. These days, more and more users prefer upgrades to be sent by email (usually as self-extracting archives of programs or disk images).

It is understandable that, in the era of a network communication, having permanently write-protected floppies piling up on your desk can be annoying. On the other hand, having your anti-virus program (or the latest update) sitting on a machine which has crashed after a 'simple' infection and is now inaccessible is not much help either. Having a hard copy of an anti-virus program is still a good idea, even if reserved for less frequent updates (e.g. quarterly).

Unfortunately, posting regular updates to users no longer suffices. With the number of viruses still growing (in the case of macro viruses, at an exorbitant speed), giving users access to daily updates is becoming a necessity for those who want to stay in business.

Most anti-virus companies currently use a mixture of distribution methods. The favoured ones are largely due to existing business practice, cost-effectiveness, and the perception of users' needs and security requirements. Today, all vendors obey the rule 'the customer is always right' – now, customers are telling producers how they want their updates to be delivered, and how often. Even the offering of the latest 'still-hot' version is not enough to satisfy all expectations.

When choosing a product, a user must be satisfied by the manner in which it can be kept reliably up to date. Soon, those programs which are able to find the relevant developer's site, download their own upgrades, and distribute and install themselves through users' networks will be the clear winners.

Jakub Kaminski, Technical Editor

NEWS

Solomon – 1; McAfee – 0

In the latest confrontation in a long-running battle between developers *Dr Solomon's* and *McAfee*, the UK Advertising Standards Authority (ASA) has upheld a complaint by *Solomon's* over an ad campaign run on both sides of the Atlantic. *McAfee's* ad for its anti-virus software led with the headline: 'The Number One Choice Worldwide. No Wonder The Doctor's Left Town'.

A statement from the ASA reads: 'The authority understood that Dr Solomon still played an active role in the company, as a director and a consultant, and concluded that the advertisement was misleading and denigratory. It understood that there were more up-to-date market share figures and considered these claims too old to be considered accurate.'

McAfee, it is claimed was unable to show the ASA any incontrovertible proof as to the accuracy of the market share it claimed for *Dr Solomon's*. Although the campaign is no longer being run, *McAfee* is reported still to be using the market share figures in current advertisements.

Recently, *McAfee* accused *Dr Solomon's* of using a 'cheat' mode in its software to give it better results in reviews [*see VB, May 1997, p.3*] ■

A Growing Trend

McAfee has also been the target of yet another anti-virus software developer. *Trend Micro Devices* is now suing both *McAfee* and *Symantec* for infringement of patents issued earlier this year on virus detection techniques used for data carried over the Internet, email, and groupware.

Trend lodged its suit in the US District Court for Northern California, and specifically cites *McAfee's WebShield* and *GroupShield*, and *Symantec's NAV for Email Gateways*. *Trend* is seeking damages, including treble damages for any 'wilful' infringement, and a permanent injunction preventing both *McAfee* and *Symantec* from further development and/or sale of any of the products involved in the suit.

Robert Lowe, speaking for *Trend*, said: 'We are confident the court will uphold the conclusions of the US Patent Office and take the reasonable actions we request to stop the ongoing infringement … We intend to vigorously protect our intellectual property.'

This lawsuit is the second recent claim against *McAfee*, which was last month sued by *Symantec* for infringement of copyright with respect to *McAfee's PC Medic* ■

Information on all of the above stories can be found at one of the following Web sites:
http://www.symantec.com; http://www.mcafee.com;
http://www.trendmicro.com; http://www.drsolomon.com

Prevalence Table – April 1997

Virus	Type	Incidents	Reports
Concept	Macro	49	14.5%
AntiEXE	Boot	34	10.1%
Form.A	Boot	28	8.3%
AntiCMOS.A	Boot	26	7.7%
Cap	Macro	24	7.1%
NPad	Macro	24	7.1%
MDMA	Macro	14	4.1%
Parity_Boot	Boot	12	3.6%
Wazzu	Macro	10	3.0%
Laroux	Macro	8	2.4%
NYB	Boot	8	2.4%
Ripper	Boot	7	2.1%
Empire.Monkey.A	Boot	6	1.8%
Showoff	Macro	6	1.8%
Sampo	Boot	5	1.5%
Colors	Macro	4	1.2%
Empire.Monkey.B	Boot	4	1.2%
EXEBug	Boot	4	1.2%
Johnny	Macro	4	1.2%
Telefonica	Multi	4	1.2%
WelcomB	Boot	4	1.2%
Appder	Macro	3	0.9%
Da'Boys	Boot	3	0.9%
Jumper.B	Boot	3	0.9%
OneHalf.3544	Multi	3	0.9%
Cascade.1701	File	2	0.6%
Divina	Macro	2	0.6%
Goldfish	Macro	2	0.6%
Hybrid	Macro	2	0.6%
Lunch	Macro	2	0.6%
Manzon	File	2	0.6%
Sharefun	Macro	2	0.6%
Tai-Pan.438	File	2	0.6%
Others [1]		36	15.0%
Total		338	100%

[1] The Prevalence table includes one report each of AntiCMOS.B, Assistant, Bandung, Beryllium, Bye, Cascade.1704, Chinese_Fish, Clock, Cmp.4096, Die_Hard, DZT.B, Form.B, Hassle, Imposter, Impulse, Jumper.A, Junkie, Kaczor, Kompu, Leandro, Nightfall.B, NF, Nuclear.J, Outlaw, Quandary, Rainbow, Sack, StealthBoot.C, Stoned.Angelina, Stoned.Manitoba, Tai-Pan.666, Trackswap, TVPO.3783, Virogen.PinWorm, and WBoot.

IBM PC VIRUSES (UPDATE)

The following is a list of updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 15 May 1997. Each entry consists of the virus name, its aliases (if any) and the virus type. This is followed by a short description (if available) and a 24-byte hexadecimal search pattern to detect the presence of the virus with a disk utility or a dedicated scanner which contains a user-updatable pattern library.

Type Codes

C	Infects COM files	**M**	Infects Master Boot Sector (Track 0, Head 0, Sector 1)
D	Infects DOS Boot Sector (logical sector 0 on disk)	**N**	Not memory-resident
E	Infects EXE files	**P**	Companion virus
L	Link virus	**R**	Memory-resident after infection

Aiwed.678
ER: An appending, encrypted, 678-byte virus. At 1:00 am or 8:00 am the payload generates a sound.

```
Aiwed.678           E867 00F8 B8AD DECD 2172 4BE8 7402 0E07 32C0 B91C 00BF A602
```

Andromeda.1024D
ER: A stealth, appending variant of this 1024-byte virus family.

```
Andromeda.1024D     B942 032E 8A04 32C4 2E88 0446 3BF1 75F3 C3B4 2CCD 2102 C402
```

Andromeda.1024F
ER: An appending, 1024-byte virus containing the texts '????????EXE' and 'AXE'. Infected files end with the character 'J' (48h). The following template also detects variant E.

```
Andromeda.1024F     06BE DFAF B430 CD21 81FF C3C3 751C 8CCB 2EA1 1803 2BD8 2E89
```

AntiAVP.959
CN: This appending, 959-byte direct infector targets *KAMI Associates' AVP*. It contains the texts 'AVp.SeT', 'KRN386.AVB', 'kRn386.aVb', '*.cOm', 'c:\DoS\fORmaT.cOM', '[AVP-Aids, Tcp / 29A]', 'AVP Aids!', 'aids' and 'by Tcp'.

```
AntiAVP.959         B802 4299 33C9 CD21 B440 8BD5 B9BF 03CD 21B8 0042 9933 C9CD
```

AntiAVP.1235
CR: An appending, 1235-byte virus targeting *AVP*. It contains the text '[AntiCARO, by Mister Sandman/ 29A]', and a message starting: 'Please note: the name of this virus is [AntiCARO] written by Mister Sandman of 29A...'. It also contains the texts 'avp.set', 'BIZATCH.AVB', 'bizatch.avb', 'Bizatch_', ' _Page_C _Header _Seek _Read', and '_decode'.

```
AntiAVP.1235        2689 4515 B440 B9D3 04BA 0000 CD21 B43E CD21 5826 8845 04EB
```

Antiheuristica.672
CN: An encrypted, appending, 672-byte direct infector containing the texts 'θVirus Anti-Heur!stica v. 2.0 (c) 1995 Spain.θ' and 'c:\dos\*.com'. There are only two (16-byte) possible templates:

```
AntiHeuristica.672  BD?? ??B8 0325 8D96 9103 CD21 CCE8 7E02
AntiHeuristica.672  1801 8B86 9403 B93C 0131 0446 46E2 FAC3
```

ARCS.1194
CN: An appending, 1194-byte, fast, direct infector with the texts '*.com', '????????COM', 'GSOP' and 'ARCS'.

```
ARCS.1194           8B1F B9AA 0490 B440 5A52 81EA BC00 CD21 5B53 81EB A700 8B1F
```

Arequipa.1994
CER: A stealth, encrypted, 1994-byte virus containing the texts 'Error, memory 1F8E:07A2 hardware internal ...' and 'SCAN.EXETBSCAN.EXETBAV.EXEMSAV.EXE'.

```
Arequipa.1994       8B05 3307 8905 83C7 02E2 F5C3 E800 005F 8BDF 81C7 1500 B949
```

AstronSolar.1056
CN: An appending, 1056-byte virus which contains the texts 'Astron.Solar by 1996-96 Inc.' and '*.com'. The virus reinfects files which have already been infected.

```
AstronSolar.1056    8ED8 B920 04B4 40CD 21B4 3ECD 2158 8ED8 5AB8 0143 B901 00CD
```

BabyC.128
EN: A companion, 128-byte virus containing the text '*.EXE'.

```
BabyC.128           E815 0072 0F93 B440 B980 00BA 0001 CD21 B43E CD21 B44F EBDF
```

BlackMonday.928
CR: An appending, 928-byte virus. Its payload tries to log the user out of the *NetWare* system (4.0 or higher or Alloy network) but because of a bug this does not work. The virus reinfects infected files.

```
BlackMonday.928     0189 169B 00BA 0000 B800 40B9 A003 CD21 7229 B800 42BA 0000
```

Cancerbero.1864
CER: An encrypted, appending, 1864-byte virus containing the texts 'Disk Full. Press any key to continue', 'This program was written in Argentina', 'Copyright 1994-1995 Cancerbero [DAN]', 'C:CHKLIST.MS', 'C:CHKLIST.CPS', 'C:ZZ##.IM', 'anti-vir.dat', 'ANTI-VIR.DAT' and 'Greetings to all [DAN] members'. The payload overwrites 65535 sectors on drive C.

```
Cancerbero.1864     AC32 C2D0 C8F6 D0C0 C005 AAB4 02CD 17FE C2E2 ED3D BDAA 2AAB
```

Cannabis.1029
ER: An appending, 1029-byte virus containing the text 'No! Cannabis...'.

```
Cannabis.1029       33D2 B905 04B4 40CD 2180 3EB4 0301 7408 B000 E874 00EB 0A90
```

CivilWar.438
CN: An overwriting, 438-byte virus containing the texts '*.com', 'File corruption error' and 'Civil War My hands are tied, For all I've seen has changed my mind, But still the wars go on as the years go by, With no love of God or human rights, 'Cause all these dreams are swept aside, By bloody hands of the hypnotized, Who carry the cross off homicide, And history bears the scars of our civil wars'.

```
CivilWar.438        B440 B9B6 01BA 0001 CD21 7223 B457 B001 5A59 80E1 C080 C92C
```

CivilWar.440 **CN:** A minor 440-variant of CivilWar.438, containing exactly the same text.

```
CivilWar.440        B440 B9B8 01BA 0001 CD21 7225 B457 B001 5A59 80E1 C080 C92C
```

Desert.641 **CN:** An appending, 641-byte direct infector containing the text ':*.COM', '.COM' and '????????COM'. The payload triggers when all files in the current and the C:\DOS directories are infected, and overwrites the contents of the first physical hard disk. Infected files have their time-stamps set to 62 seconds.

```
Desert.641          B440 B981 028B D681 EA02 02CD 2173 03E9 8D00 3D81 0274 03E9
```

Devastator.301 **CN:** An overwriting, 301-byte, direct infector containing the texts '*.COM', 'Devastator XI', 'Now includes destructor code! 1994 / 12-21-94'. The payload overwrites contents of a first physical hard disk.

```
Devastator.301      BB1A 0143 8A16 0301 3017 81FB 2C02 75F3 C606 0301 041E 33C9
```

Hungry.633 **CN:** An appending, 633-byte virus containing the texts 'Virus Ver 1.01a Copyright (c) 1994 by Hungry Software', 'AMI', '*.COM', and '????????COM'. All infected files have their time-stamps set to 62 seconds. The payload corrupts the CMOS data.

```
Hungry.633          8BD6 B440 2E8B 9C3D 00B9 7902 0E1F CD21 2E8B 8476 002D 0300
```

Jovial.506 **CN:** An appending, 506-byte virus containing the texts 'JOVIAL KINDNESS BY yOUNG aDULT mALE', 'HI MOM!!!! ', '[NOP/HLT ENGINE 1.0]' and '*.COM'. The word 484Eh ('NH') is at offset 0003h in infected files.

```
Jovial.506          EA02 CCE8 B700 B440 B9FA 018D 9605 01CC 90E8 8A00 EB04 90EB
```

KVS.1942 **CER:** An appending (exe) and prepending (com), 1942-byte virus containing the plain-text strings 'KieViruSoft (c) Ver1.0' and 'Ver1.0', as well as the encrypted text 'Take Care of SoftWare ... KieViruSoft Data Product (c) 1994 .'.

```
KVS.1942            BA96 0781 C296 0781 C22C 01B1 04D3 EA42 CD21 2EFF 36E0 041F
```

Lobotomy.966 **CN:** An appending, 966-byte virus containing the encrypted text '*.com'.

```
Lobotomy.966        4033 D2CD 21B4 40B9 C603 90BA 0001 CD21 B800 4233 D233 C9CD
```

Npox.611B **CR:** An appending, 611-byte virus containing the text 'Rock Steady/NuKE'. The payload triggers on the 24th of a month and includes formatting the first hard disk. Infected files' time-stamps are set to 58 seconds.

```
Npox.611B           4E75 4B45 5D55 3E8A 865F 02B9 3A02 2E30 4600 F6D0 45E2 F7C3
```

Overdoze.573 **CR:** An appending, 573-byte virus containing the texts '[Overdoze] (c) 1994 The Unforgiven/Immortal Riot', and 'Dorked with by the EVG/Executioner'. Infected files have byte 56h ('V') at offset 0003h and their time-stamps set to 2 seconds.

```
Overdoze.573        8BEE 2BC0 80CC 660C 66CD 2181 FB66 6674 6A0E 1F6A FF5B B44A
```

Overdoze.580C **CR:** An appending, 580-byte virus containing the texts '[Overdoze] (c) 1994 The Unforgiven/Immortal Riot', and 'Dorked with by the EVG/Executioner'. Infected files have byte 56h ('V') at offset 0003h.

```
Overdoze.580C       B410 80C4 1880 C418 B944 02CD 212B C00D 0042 2BC9 CD21 2BC9
```

Overdoze.593 **CR:** An appending, 593-byte virus containing the text '[Overdoze] (c) 1994 The Unforgiven/Immortal Riot Dorked with by the EVG/Executioner'. Infected files have byte 56h ('V') at offset 0003h and their time-stamps set to 2 seconds.

```
Overdoze.593        8BEE B066 B466 CD21 81FB 6666 746B 0E1F B44A BBA5 2581 C35A
```

Pamyat.2000C **CEN:** An encrypted, appending, 2000-byte, fast, direct infector containing the texts '*.COM', '*.EXE', 'PATH=COMSPEC=OBSHCHESTVO=', 'AIDSTEST.EXE', ' AIDSTEST!' and '10-4-1995. Version 3a'.

```
Pamyat.2000C        E800 005B 83EB 03B9 D007 BE00 000E 1FB0 D130 401C C0C8 04FE
```

PSMPC.313 **CN:** An appending, 313-byte virus which infects one file at a time. It contains the texts '[MPC]', '[SHY_KOO]', '[Walt Whittman]' and '*.com'.

```
PSMPC.313           33C9 99CD 21B4 408D 9603 01B9 3901 CD21 B801 578B 8E56 028B
```

Smile.1113 **ER:** An appending, 1113-byte virus containing the texts 'Access denied' and 'Smile Virus'. The payload includes a number of screen effects.

```
Smile.1113          B440 B959 04BA 0000 9CFF 1EFF 03B8 0157 8B0E FB03 8B16 FD03
```

Spanska.1500 **CEN:** An appending, 1500-byte virus containing the texts 'Mars Land, by Spanska(coding a virus can be creative)', '*.*', '*.C*', '*.E*' and '..'.

```
Spanska.1500        AAC3 8A96 2601 B9AC 058D B63F 018B FEAC 9032 C2E8 EAFF E2F7
```

Tease.1362 **CER:** A stealth, encrypted, appending, 1362-byte virus containing the texts 'TEA TOAST AND TITANIC TITTIES!', 'c:\dos\doskey.com', '[CONFUSION MELTDOWN]' and '(c) Pottie Rottie, Sweden 1994*.com'. Infected files' time-stamps are set to 58 seconds.

```
Tease.1362          3E8B 963D 068D B60D 01B9 9802 3114 4646 E2FA C3FF FFFF FFFF
```

Tiny.200 **CN:** A prepending, encrypted, 200-byte, direct, fast infector containing the text '*.*'.

```
Tiny.200            5006 1E07 8A26 A701 8BFE AC32 C4AA E2FA 0758 C3?? E80F 00B4
```

Torero.1427 **CR:** An appending, 1427-byte virus containing the text '[Torero Ç:-) by Mister Sandman/29A]' and ';)This program requires Microsoft Windows.'

```
Torero.1427         3D60 EA77 3D50 E81C 01B4 40B9 9305 BA00 00E8 4700 582D 0300
```

WarCannibal.238 **CEN:** An overwriting, 238-byte, direct infector containing the texts '\Var Cannibal Animal..' and '*.*'. After infecting all suitable files in the current directory, the message: 'Incorrect DOS version' is displayed.

```
WarCannibal.238     B440 B9EE 00BA 0001 CD21 B801 572E 8B0E 9600 2E8B 1698 00CD
```

INSIGHT

Aubrey-Jones: The First Crusade?

David Aubrey-Jones – father, adventurer, and software developer. A Londoner born and bred, Jones was still young when he escaped the city smog to live in the open spaces of the English countryside. Well, almost… as *Reflex Magnetics'* Technical Director, he is still in the city every working day, but professes himself satisfied with this compromise.

Jones' original career path was in nutrition, in which he has a doctorate from the University of Leeds – it was during his research for his thesis that he first came into contact with computers: 'The first computer I ever used,' he reminisced, 'was a PDP 8 which was purchased by our university department whilst I was doing research for my PhD. I started to use it extensively for calculating my research results and soon became the expert.

'At the time I thought it was marvellous, with its 4KB of RAM memory and programming on paper tape. It didn't even have the luxury of a VDU screen, and all instructions were input on a very noisy teletype machine!'

From Food to Fortran

He soon 'graduated' to the university mainframe, an ICL computer running the George operating system. During this time, he gained experience of Fortran, and worked with punch cards: 'CPU time was much in demand, and limited: it was not unusual for the computer to grind to a halt by mid-morning. It would get slower and slower, and eventually would lose anything you were currently editing. Consequently, I started to do most of my work in the evening when the computer would run ten times faster.

'Apart from editing, all main jobs, such as running a program, were normally done overnight on a batch basis. If there were any mistakes or bugs in your program, you didn't find out until you got the print-out the following day. This certainly taught me to be careful while programming and try to get things right the first time; if you didn't, a program could take months to get right and debug.'

It was during this period that PCs first appeared, and Jones, like many others, was soon bitten by the computer bug. He spent much time gazing enviously at such unaffordable consumables as *Apple* computers and *Commodore PETS* – and then came *Sinclair*, with the *ZX80* and, later, *ZX81*.

'The desire to have my own computer was overwhelming,' he said, 'and I succumbed. At first, it was wonderful; mine to use as I pleased; no more waiting until next day to discover yet another bug in my program.'

It was not long, however, before the limited speed of his new acquisition pushed him into learning Z80 Assembler, and he was quickly 'hooked': 'The main demand at this time was for games programmers,' he recalled, 'and there were very few who had really mastered the art of Assembler programming (which there still are). I wrote my first commercial game for a small UK company – this led to many offers.'

The first game he sold commercially was called 'Cowboy Shootout'. This was one of the very first games for the *Sinclair Spectrum*, and Jones subsequently went on to do the official 'Galaxians' conversion to the *Spectrum* for *Atari*: 'To make a faithful copy of the original arcade,' he said, 'and get it on a computer with such limited processing power was a real challenge.'

Jones was, however, interested in far more than just games: he was fascinated by the concept of software piracy – could not a method be developed to counter this? Although friends and colleagues said it was impossible, Jones was determined to resolve the dilemma, and set to work with a will. After considerable research, he and a friend came up with a solution which was adopted with great success by the burgeoning computer games industry.

'Over time,' he said, 'hackers started to try to break through the protection code. This led to a cycle of continual development, and increased sophistication, as the battle progressed. We soon developed automatic layered encryption systems using hundreds of separate layers. In many ways, it was similar to today's fight with the virus writers.'

All this happened in the early 1980s, around the time of the first *IBM* PC. The copy protection program, *Speedlock*, was first used on the *Sinclair Spectrum* when all program loading was on tape. One function apart from protection was to speed up load times: versions were written for disk, and for nearly all the most popular home computers.

Here a Virus…

Jones' first exposure to a computer virus was on the *Atari ST* and the *Commodore Amiga*: 'One of the first I saw,' he remembered, 'was one which reversed mouse direction. A friend became infected with it and spent hours taking his mouse apart and examining his hardware to find the fault. Then, during the copy-protection work, I started to receive an increasing number of programs for duplication mastering that were virus-infected. It therefore became essential that these viruses were detected, and I began to analyse them.'

The first virus analysis he read was an article on the Stoned and Brain viruses: before the days of *Virus Bulletin*, this was published in a UK weekly computer newspaper and written by one Alan Solomon.

David Aubrey Jones: mining for rocks, data, and software solutions.

A Glance at the Future

‘A couple of years ago,’ commented Jones, ‘I thought that the virus problem was likely to decrease in the future. Boot sector infectors were by far the most common viruses in the wild, and more advanced operating systems such as *Windows NT* prevented them propagating. This picture has changed with the advent of the macro virus, and I believe the future is now far less certain. In fact, I think there is now a major paradigm shift occurring, with the potentially deadly combination of macro viruses and the Internet.

‘Macro viruses are simple and fast to write, requiring very little expertise, and their numbers are now increasing faster and faster. We may be in a situation in a matter of months where we are receiving not just tens but hundreds of new macro viruses per week. If this is coupled with very rapid distribution via email over the Internet, we may be in a different ball game. Anti-virus methods may need a rethink.’

In his view, the main advantage of virus-specific detection is that it has been found the most reliable; not necessarily at detecting viruses, but in terms of producing few false alarms. Jones feels virus-specific detection is not the cure to the problem, but concedes that it is now synonymous with all anti-virus measures: ‘There is a place for specific detection,’ he elaborated, ‘but one needs to consider why we are doing it, and if there are other ways of achieving the same ends.

‘The big problem I foresee is, if you only scan for known viruses, how do you keep it up to date? It is now common practice to perform updates quarterly or monthly. If new macro viruses can be written and then spread in a matter of minutes on the Internet, what level of updates are appropriate? Weekly, daily, hourly or every few minutes?

‘Even if you can receive and deploy updates at this rate, there are other problems. For instance, how long will it take for the anti-virus industry to become aware of a new virus? If it comes straight to us, fine, but if it first infects customers? And there may be dozens or even hundreds of cases like this per day.

‘It seems obvious to me that we need to develop other techniques as a matter of urgency. Heuristic scanning will help, but I don’t believe it is the full answer. I think there is potential in behaviour monitoring and blocking that has never been fully realised. It has been dismissed by many due to a high level of alarms, but work that I have done at *Reflex Magnetics* has shown that it doesn’t have to suffer from this problem.’

Heuristic scanning has many advantages over virus-specific scanning, opined Jones, in that it can detect many new viruses: ‘It has tended to suffer from a poor reputation, as some early products using this technique were more likely to produce false alarms. It will, though, I believe, soon be an essential part of macro virus scanning.’

Personal Points

Jones is pleased with his career path up to now – he very much enjoys the development side of his work, and has broadened his interests into other areas of security, such as encryption and Java: ‘I find all aspects of IT security fascinating,’ he explained, ‘and a tremendous challenge. At *Reflex* we have a great team, and you can expect some interesting products from us in the future.’

A family man as well as a true professional, Jones met his wife Lynn whilst doing his doctorate at Leeds University, and has been married for several years. The couple has three children: Tristan, age 9; Harriet, 7; and Dominic, 5.

‘It is wonderful to watch them discovering the computer,’ said the proud father. ‘If only I had had the sort of computer that they have today, with all its multimedia capabilities! All my children love using computers. They particularly like being creative, developing graphics, stories and movies.

‘Tristan is the only one doing any real programming at present, but I am looking forward to introducing the other two when they are a little bit older. My wife, Lynn, is a nurse and health visitor, and she has recently started some work again now that the children are all at school.’

As if this busy life were not enough, Jones also admits to a passion for rocks: ‘In many ways I am like my namesake, Indiana Jones. At weekends you might find me searching for precious metals and minerals, otherwise known as rockhounding. It certainly sharpens my data mining!

‘When not rockhounding, I am often hiking or camping, pastimes that have become popular with the whole family. I am also interested in exotic plants, and have an unusual collection. My other passion is travelling, something which I share with Lynn, although we currently have little spare time to do much. One of my favourite parts of the world is the North American deserts and mountains.’

Certainly, this man’s life is full of all the things about which he is most passionate, and this passion comes across in everything he does. David Aubrey-Jones – father, adventurer and software developer. What will be next on his list?

VIRUS ANALYSIS 1

Russel: A Wily Hare with Three Burrows

Dr Cai-Gong Qin

'A wily hare has three burrows', as the saying goes. Virus writers have been racking their brains to enable them to make the life of the anti-virus researchers more difficult, by creating many complicated burrows – some of the typical examples are polymorphism, stealth, multi-encryption and anti-anti-virus techniques.

The DOS virus Russel is one of those wily 'hares'. It is a parasitic virus which appends itself to executable files, increasing their size by 3072 bytes. It is a polymorphic, multiply-encrypted virus with stealth capabilities and some anti-anti-virus techniques.

Installation

When an infected file is executed, the virus is activated. Russel uses two-level encryption. Accordingly, it contains a two-level decryption loop. The first loop, at the beginning of the virus, decrypts C00h bytes of code. The newly-decrypted code immediately follows the first decryption loop, and, logically enough, also contains the second decryption loop.

This loop then takes control and decrypts the main part (B26h bytes) of the virus code. Next, the virus checks which version of DOS is active. If it is lower than version 5, the virus will show no interest in going further and gives way to the host program. Should it be DOS version 5 or above, the virus sets off the TrapFlag (TF) bit in the flags register to prevent it from being single-stepped.

Like most memory-resident viruses, Russel makes an 'Are You There?' call via Int 21h, function 20h with DX=6543h to check whether or not it is already memory-resident. If so, this copy simply exits, returning control to the host program.

Russel also uses some 'anti-anti-virus' measures. As we know, *MS-DOS* bundles VSAFE as part of its virus protection. Russel uninstalls *PC Tools v8+* and VSAFE before going memory-resident.

This virus stays resident in a way that is slightly different from the usual techniques. It tries to reside in the upper memory blocks (UMB). First of all, the virus changes the UMB link state to add the UMBs to the DOS memory chain. Then it sets the memory allocation strategy to 'best fit, try high then low memory' before creating a new Memory Control Block (MCB) of 100h paragraphs.

If this call fails to create the required space in the UMBs, Russell falls back on the conventional method of modifying the size of the current MCB to steal the required 100h paragraphs. Either way, the virus marks the word at offset 1 of the created or modified MCB with 0008h so that the resident part will be regarded as a system program allocated by DOS. Finally, the virus copies itself into the allocated memory block, and the resident component is renamed 'SC' (presumably standing for 'scanning').

Before returning control to the host program, the virus hooks Int 21h and saves the address of the original Int 21h handler to the double words at offset 0055h from the beginning of the hooked Int 21h handler and at 0000:03C4h in memory, respectively; the latter one intercepts Int F1h indirectly.

The Interrupt Handler

The Interrupt 21h handler is the most complicated part of the virus. It consists of a variety of routines for stealth, polymorphism and infection. In addition to the 'Are You There?' call (AH=20h, DX=6543h), it also intercepts the DOS functions 3Eh (Close File) and 4Bh (Load and Execute). During infection, the virus hooks Interrupt 24h (the critical error handler) to suppress error messages.

> *"when a program is loaded and executed through DOS function 4Bh, the virus will be unconditionally activated"*

Before attacking a potential target, the virus checks the extension of the target's filename to see if it is executable. The virus does not infect any files with names matching *nd.* or *an.*; therefore, COMMAND.COM remains intact. Further, it does not infect any files with a size greater than or equal to 60,000 bytes.

After verification of its target, Russel unsets the file's Read attributes. It copies the first 28 bytes of the file into memory to check if the file has a COM or EXE structure, and whether or not it is already infected. Those files whose first word is not the EXE signature 'MZ' or 'ZM' are assumed to be COMs.

COM files infected by this virus always start with two fixed bytes: F8h and E9h. The virus uses this feature to prevent infected COM files from being infected again. Similarly, an EXE file will be checked against its header to see if it is infected by this virus.

When a file is opened or created through the DOS function 3Ch (Create file with handle), 3Dh (Open file with handle), 5Bh (Create new file) or 6Ch (Extended Open/Create), the virus will save the file handle to the word at offset 0C02h in its Int 21h handler. Afterwards, when an infectable file is about to be closed using the DOS function 3Eh, the virus first checks the current file handle against that saved to see if the file to be closed is the most recently opened or created. If it is, the virus closes the file and starts infecting it; otherwise, Russel leaves the file alone.

So, if several files are currently open, only the last one opened is at risk of infection; the others are not. However, when a program is loaded and executed through DOS function 4Bh, the virus will be unconditionally activated.

The engine for Russel's polymorphism is included within the hooked Int 21h handler. It inserts at random a set of 'garbage' codes (for example 90h, CCh, F5h, F8h, F9h, FCh and FDh) into the decryption loop at the beginning of the virus, based on reading the value at 0040:006Ch (the number of ticks of the system timer).

In addition, the first-level decryption loop is made polymorphic by use of a combination of XOR, ADD and SUB instructions, while the second one uses the ROL (Rotate left) instruction only.

On exit to the host program, the virus restores the host's original attributes, time and date, as well as unhooking for Int 24h.

Conclusions

Russel is a parasitic virus which employs many tricks to challenge anti-virus researchers. It is polymorphic, has stealth capabilities, and multiply-encrypted and anti-anti-virus. Files are infected while they are executed or copied. It demonstrates that some virus writers have become more crafty, and fiercer, in the virus version of 'Star Wars'.

Russel

Aliases: None known.

Type: Multiply-encrypted, polymorphic, memory-resident.

Infection: All executable files; however, there are some exceptions (see text for details).

Self-recognition in Memory:

Check data in segment of Int 21h handler.

Hex Pattern in Memory:

```
E984 019E 1016 0132 C0CF E486
00F6 D980 E107 5188 0EE0 0BBE
```

Intercepts: Interrupt 21h, functions 3Eh and 4Bh, for infection, polymorphism, and stealth.

Trigger: None.

Removal: Under clean system conditions, identify and replace infected files.

VIRUS ANALYSIS 2

Deadly NightShade

Martin Skilling

The release of *Microsoft's Office 97* brought a new challenge for the virus writers. Here was a new environment for which no viruses currently existed, which even contained built-in virus protection, designed specifically to prevent the upgrading of the old *Word 6.0/95* macro viruses to this new platform.

This protection, however, is far from complete: it entirely misses many of the older *Word* macro viruses. These are successfully translated, by *Office 97*, into its new macro environment: the thrillingly-named Visual Basic for Applications version 5 (VBA5).

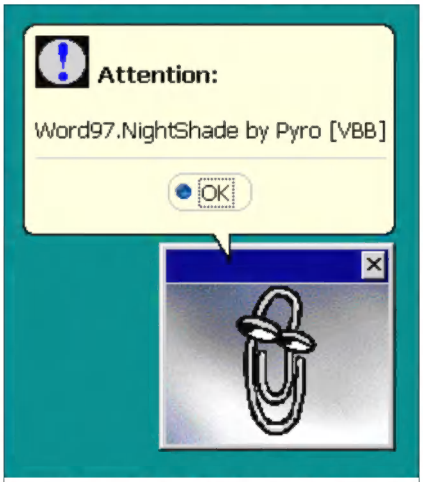
A New Genre

NightShade is one of the first of a new type of macro virus; namely, those written specifically in VBA5 and which only spread under *Office 97*. Although in theory the same VBA5 environment is used by all the major components of *Office 97* (*Word*, *Excel*, *PowerPoint* and also, in the Professional Edition, *Access*), the current *Office 97* macro viruses only target one component, which in this case is *Word 97*. It can only be a matter of time before someone writes a virus designed to target specifically one of the other components.

This particular infector is a small, self-contained, single-macro virus. Assuming AutoMacros have not been turned off, *Word* executes the macro AutoClose whenever a document is closed – this is where the virus seizes control.

In Operation

Some amount of thought has gone into making the virus' operation as silent as possible so as to allow it to spread far and wide. The first thing NightShade does is attempt to ensure that its execution is invisible, even if errors occur. It turns off not only screen updating, but also the alert boxes which *Word* puts up when it is executing something it considers to be dangerous.



Attention:

Word97.NightShade by Pyro [VBB]

Figure 1: NightShade uses *Office Assistant* to announce its presence.

Next, it does what it can to ensure that it gets to run as often as possible, by telling *Word 97* to execute AutoMacros and by turning off the built-in virus protection.

This means that the user only has to allow the macros to be enabled once for the virus to install itself properly. Thereafter, *Word 97* will silently allow virus-infected documents to be loaded without the appearance of any warning message onscreen. This, of course, also lays *Word 97* open to infection from other viruses. Being able to turn off the virus protection from a program can easily be considered a major fault in the product.

Self-recognition

The next section of the viral code determines whether or not the active (i.e. current) document is already infected by going through any associated Visual Basic project components and comparing their name with 'NightShade'.

> *"(NightShade) demonstrably shows that native* Office 97 *viruses are not just a theoretical possibility"*

It then goes through an almost identical loop, but this time compares the Visual Basic project components found in the global environment. NightShade makes no assumptions about the name of the file used to store the global environment and makes good use of the Visual Basic object hierarchy. It should, therefore, have no trouble spreading under any non-English versions of *Word 97*, and also under non-standard installations.

Both the document and the global environment have to be checked for infections in order for NightShade to function properly, because exactly the same macro executes from infected documents (i.e. templates) to infect the global environment as executes from the environment to infect documents.

Once it knows which area is not currently infected, it copies the macro code into the uninfected area, making the valid assumption that the other area will already be infected and will therefore contain the viral macro to copy.

If it is infecting a document, it saves that document, making sure that the file type is set to Template, so that the next time it is loaded into *Word 97* the macro is recognized. If it is the global environment that is being infected, the prompt before saving any alterations is turned off.

Trigger Routines

By this point in its execution, the virus has performed its most important task; namely, that of replication (if this was necessary). Now it can proceed to carry out other, more frivolous, activities. There is a one-in-seven chance that NightShade will attempt to announce its presence using the new feature of *Office 97*, which some would call cute, and others irritating – the *Office Assistant*.

If the *Assistant* has been installed, it will appear at this point, complete with balloon text reading 'Attention: Word97.NightShade by Pyro [VBB]'. (See Figure 1, p.9)

If the *Assistant* has not been installed, the virus' attempted use will cause an error. However, this will not be noticeable, because the virus traps all errors and forces a jump straight to the end of its code. This means that, in this eventuality, its second payload may be skipped.

This second payload is somewhat malicious in that it password-protects the document. If the date is Friday the thirteenth, and the document is not already protected thus, the virus effects this, using the password 'NightShade'.

This action does not take place if this is the initial infection of the document, or if no edits were made to the previously-infected document when it was closed. This is because simply setting a password on a document does not in itself mark the document as dirty; therefore, *Word 97* will not save it again before closing it.

The virus' final action is to re-enable the display of alerts in an effort to ensure that the user does not notice anything amiss in the way that *Word 97* works.

Conclusions

Apart from its use of the *Office Assistant*, WM97.NightShade does nothing that a *Word 6/95* macro virus could not do. The main reason for particular interest in this virus is that it demonstrably shows that native *Office 97* viruses are not just a theoretical possibility – they exist now! In the future, as users begin exchanging *Office 97* documents regularly, we can expect to see many more viruses targeting this environment.

NightShade

Aliases:	None known.
Infects:	Microsoft Word 97 documents and templates.
Self-recognition:	
	Checks for the presence of a Visual Basic project component called 'NightShade'.
Hex Pattern:	

	`3620 4173 208A EA27 3133 EE25`
	`E04C 4755 4880 6173 5061 7373`

Trigger:	Friday the thirteenth.
Payload:	Password-protects document with the password 'NightShade'.

VIRUS ANALYSIS 3

Olivia

Péter Szor
Data Fellows

The Olivia virus was found in the wild in April 1997: first reports in the wild came from three different countries – Taiwan, Poland and Hungary. In a now-familiar fashion, it was distributed over the Internet, like so many other viruses before it.

Several infected files and droppers have been uploaded to a Taiwanese FTP site. One of these, RAR25C.EXE, claimed to be a new beta version of the popular archive utility, RAR, from Russia. When this self-extracting executable is run, it unpacks two other files, RAR.CFG and RAR.EXE. RAR.EXE is a dropper for Olivia.2374.

When RAR.EXE is executed by the user, it displays the error message: 'Not enough memory. Program aborted.' The user will therefore not suspect that anything is amiss, and is bound to think: 'Just the usual beta again, let's wait until the fix comes.' But the virus is already resident.

The popularity of the Internet appears to be changing the face of the virus problem in East European countries. Prior to 1996, Hungary, for example, had problems with new viruses from Romania, Slovakia, Poland, Russia and of course from Bulgaria. Viruses sometimes entered Eastern Europe from such West European countries as Germany and Sweden. There were also many cases from the Far East, including such incidents as viruses being 'imported' from Taiwan as 'OEMs' with a dozen infected PC clones.

The Internet became big business in East European countries, where large computer networks did not exist before the early 1990s. The situation as regards piracy is also getting better, which creates a new set of problems. Private individuals are trying to get the best shareware and, of course, the latest versions available. This means that a virus writer can reach his goal easily by uploading his latest, even buggy code to some sites in a hack or beta version package.

The Olivia virus family is one of the most recent examples of this trend. Although Olivia.2734 has obviously been written by a newcomer to the field, it nevertheless shows several interesting ideas in its infection mechanism and activation routine.

Initial Infection

When an infected EXE file is executed, the decryptor takes control immediately. This is not, however, the case with COM infections. When a COM file is infected, the virus follows some of the instructions in the victim's code and calls its decryptor from there.

Olivia's first trick appears almost at the beginning of its code. The virus makes a division by zero after changing the Int 0h (Exception) vector to point into the decryptor first. This routine uses the stack frequently, as well as 286 instructions, thus creating code which is both anti-emulating and anti-debugging. The decryptor is based on a random 8-bit XOR key, but it is not polymorphic, only oligomorphic. Although some of the indexes change, it is still possible to pick up a search-string.

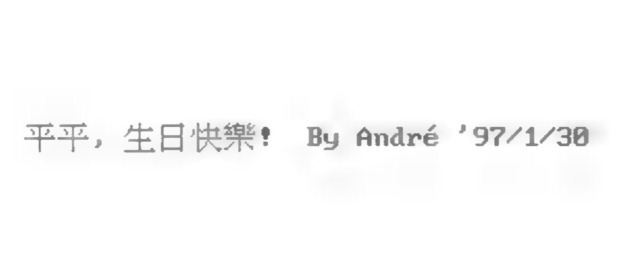
When the virus code is decrypted, Olivia disables the resident parts of *Norton AntiVirus* and *Microsoft AntiVirus*. Next, it executes an anti-emulation trap. It calls Int 11h (equipment determination interrupt) and checks that the return value in AX is not zero. In this manner, Olivia is able to detect most of the heuristic analysers which are based on emulation but which do not pay attention to this particular interrupt. The virus returns to DOS if the return value is zero.

Olivia's next action is to check the date: if it is 10 April or 23 December the virus calls its payload; otherwise it checks for its presence in memory. The 'Are you there?' call is Int 21h AX=3DA0h, BX=1980h.

If, on return from the call, the BX and CX registers are set to 1979h and 1223h respectively, the virus assumes it is already active in memory and returns control to the host program. If the BX and the CX registers do not contain these values, Olivia manipulates the MCB and copies itself to the allocated memory area. Then it hooks Int 21h. Finally, control returns to the host program.

Infection of Files

Olivia infects COM and EXE files as they are run or renamed, or as their attributes are changed. First, the virus clears the attributes of the file; then, it opens it for read. Next, it obtains the System File Table Entry of the victim and checks its time-stamp – files with time-stamps set to 60 seconds are assumed to be already infected.



平平，生日快樂！ By André '97/1/30

Figure 1: Olivia's message, displayed in traditional Chinese characters, translates as: 'Ping Ping, Happy Birthday!'

If the time-stamp is not set to 60 seconds, the virus changes the file access mode in the System File Table and checks the extension of the file there, too. If the extension is COM, the virus uses its exclude list with files named 4DOS, COMMAND, and VT, which it will not infect.

However, if the extension is EXE, the virus uses a different exclude list, which contains the names WIN, EMM386, SSCAN, TB, and CHKDSK.

Since the file WIN.COM (which launches *Windows 3.x*) has a COM extension, this check will fail, and the virus will infect that file. There is no known reason for the presence of 'VT' in the COM file exclude list, and 'SSCAN' stands for Super Scan, a local product.

If the victim has a COM extension the virus uses a special function which reads 4 bytes in a loop from the beginning of the victim and checks for E9h (JMP), EBh (JMP short), 90h (NOP), F8h (CLC), F9h (STC), FAh (CLI), FBh (STI), FCh (CLD), and FDh (STD) each time.

If one of the above instructions is found, the virus moves to the location of the next instruction. If this instruction is in the last 64 bytes of the program, the virus will not infect that file.

This means, therefore, that the virus will not infect most goat files. However, if the last instruction was not in the last 64 bytes, the virus will modify the host program at this location. More specifically, it uses the 64h (286 Push) opcode to push a word value to the stack, then executes a C3h (RET), which will give control to the virus code. This can prevent heuristic analysis if the emulator is not able to handle 286 opcodes correctly. This technique, called 'inserting', also makes disinfection more difficult.

Then Olivia modifies its decryptor, encrypts the virus body, and adds itself to the end of the COM file. The decryptor is only oligomorphic, but the virus writer is not far from writing a full polymorphic virus. Since the virus does not check the size of the COM files before infection, the infected COM files can be bigger than 64KB and will fail to execute. All these facts tell us that the virus writer is a beginner at his job.

Since the virus has stealth capabilities, the change to file size is not visible: Olivia changes the return values of Find First, Find Next functions by subtracting 2374 from the infected file size field – this is why the virus changes the file stamp to 60 seconds at the end of the file infection.

In the case of EXE files, Olivia does not pay special attention to the file structure; the result being that it will infect standard goat files.

Payload

The virus calls its payload when the date is 10 April or 23 December (the year is irrelevant). First, Olivia checks that the PC has an active hard drive, by examining the CMOS. Then it checks for an installed CD-ROM drive. If a CD-ROM is available, Olivia opens the drive and displays this message:

```
please put a love music CD into your CD-ROM
and pass any key to continue...
```

Then it waits for a keypress. If the user puts an audio CD into the CD-ROM drive and presses any key the virus closes the drive and starts to play it.

Next, it changes the video mode and displays a message onscreen which contains Chinese characters (see Figure 1). The translation of the text is: Ping Ping, Happy Birthday! The characters are traditional Chinese ones, normally used in mainland China.

Olivia then disables the keyboard, and then it clears the contents of the CMOS. Finally, it overwrites the hard drive, using memory address FFFFh:0 as the sector image.

The virus has an additional message, never displayed:

```
Olivia Virus 7.5ß By André (C)TRAN TECH United
Groups
```

Summary

Olivia.2374 shows that a virus written by a beginner can spread far and wide if some infected files are available from the Internet. People should be extremely careful in handling material downloaded from the 'net.

Those who do not take precautions are likely not only to lose data from their home PC, but also to create problems for their company. Free software does not necessarily mean virus-free software.

Olivia

Alias: CDROM.

Type: Resident, stealth, oligomorphic.

Infection: COM, EXE.

Self-recognition:

60 seconds marker in files.

Hex Pattern in Files:

```
C08E D8FF 3600 00FF 3602 0068
???? 8F06 0000 8C0E 0200
```

Hex Pattern in Memory:

```
CD16 E800 0033 C0CD 110B C075
04B4 4CCD 2144
```

Payload: Plays Audio CD, displays messages, overwrites the CMOS and the hard drive on 10 April and 23 December.

Removal: Recover affected files from backup or replace with original.

FEATURE

Through the Administrator's Eye

Phil Crewe
PERA Group

These days it is very easy to put in place a virus scanner which, on the surface at least, will scan for all known viruses (some scanners can scan for unknown ones as well), giving your machine the best protection money can buy. Most also include a monthly update service to ensure that the user stays ahead of the game.

For any reasonably technical person, having a machine without some degree of virus protection must be viewed as almost a dereliction of responsibility. There are, however, many cases that do not fall into this simple reasoning.

If you are an administrator in charge of many machines in a corporate environment, the issue is more than merely providing a degree of virus protection to a technical user. Also to be considered is usability users with a wide range of skills, scalability of protection required, and applicability to potentially multiple client platforms and to different network architecture. Further, there are the normal administration and updating headaches which come from running any piece of software on multiple machines, which are probably also separated geographically.

And this is just the technical side of the problem! Installing software on any machine brings with it a set of problems for the customer support department – virus protection is no exception. The support department will have to field the false positives generated, as well as having to manage the expectations of the users.

These users must accept that installing anti-virus software does not negate their responsibility, but merely provides a degree of safeguard for when they are using data and applications from floppy disks or public service networks. It does *not* mean they have *carte blanche* to bring in software (games or otherwise) from home simply because the virus issue may be being addressed by using a software solution.

At the User Level

Let us look at what is probably the largest sticking point of installing any virus-protection software. The software has to be almost invisible at the user level, except when a virus is detected. It must be quick to scan information, and should never get in the way of what the user is trying to do.

Ultimately, a user is employed by a company to do a particular job. Whatever that is, no software installed on a client machine should interfere with that. Software should enhance users' ability to do their work, not interfere with it.

From many users' points of view, virus-protection software is probably the greatest potential interference on a user machine. It will do nothing to enhance their ability to do the job, but might get in their way with false error messages, could slow the machine down, or even forbid them to use certain devices such as floppy disks. Solutions such as these tend to be acceptable only in certain circumstances (such as on military bases), where the issue is larger than just viruses.

Not Just for Viruses…

When anti-virus software is being considered, it should not be forgotten that installing a corporate virus protection policy may give additional benefits over and above pure virus protection. For example, it could be viewed as an opportunity to ensure that all floppy disks coming into and out of the organization are logged, thus giving the corporation better control over some of the other security issues that it could be facing.

It is also a chance to take additional control over client machines from a management perspective. For instance, it is an opportunity to audit all client machines to ensure that any fixed asset register is up to date.

An administrator might therefore use the opportunity to visit client machines to install virus protection and do a general scan on the machine, thus gaining valuable information for other items such as maintenance contracts or software update planning. If this is the case, it is also a good idea to consider, at the same time as the anti-virus software, the PC management software which is to be implemented in order to do all this.

Since any good strategy re-uses this kind of information to maximum benefit, and since most companies are currently considering their response to the millennium issue with respect to desktop machines, information gleaned from this process may be valuable at many points down the road.

On Handling Updates

It is most certainly neither desirable nor time-efficient to have to visit every desktop machine at regular intervals in order to keep an anti-virus package up to date. Therefore, another issue which must be considered when thinking about which package to implement is how the updates are to be distributed to client machines.

Software distribution packages are available as separate units: if the company has standardized on a particular software update package, it may well be that anti-virus updates can be distributed in this manner. Remember that software distribution can be as simple as a file being downloaded automatically from a file server whenever a user logs on, and need not be based on the more complicated packages available. Note, however, that such packages are specifically written to cater for large corporates and mass software distribution, so their functionality and facilities are much more complete.

Another solution is to implement a client/server solution where both clients and the file server to which they are attached have the same anti-virus package. A careful choice of package will enable the distribution of the updates directly from the server system.

Naturally, even this update process should comply with the general resolution; i.e. not to interfere with the users' work in any way. The user does not need or wish to know that the anti-virus data file has been updated, even if it is just with a dialogue box with an 'OK' button.

Easy as it is to hit that button at that point, the majority of users will find this unacceptable. Most people have a routine when powering up their machine in the morning, and additional dialogue boxes will interfere with this timing. Whatever system is being planned, always bear this golden rule in mind: 'Never disturb the user unless you have something interesting to say.'

Informing the End-user

This last issue raises the next point – how to tell users when you have something interesting to say. The only time the user should be informed is when a virus (or a suspected virus) has been found. In all other cases (no matter what they are), a message could be generated back to a systems administrator, where it can be interpreted properly, and further action implemented, if necessary.

Triggers for a message to the administrator could be a failure to load part of the software, or the fact that the data file is out of date, or that a user is doing something outside the security policy. In all cases, the administrator should be informed directly: reliance should not be placed on the user informing the administrator. Thus, another 'requirement' of an anti-virus system should be integration with the internal electronic mail or alerting procedure used in the company.

When a virus has been found, what should the user see? Primarily, the user should be prevented from infecting anything else. This is the first point at which user productivity can legitimately be interrupted. At the least, the user should be prevented from logging onto the network, if the machine itself does not lock up.

Whatever takes place, the user now needs to be told exactly what is happening. There is no point in locking the machine so the user cannot do anything, if you do not give the reason for such action. This may sound obvious, but there have been circumstances where the first a Help Desk knows about a virus problem is when several users from a department ring to say that they are completely fed up with the network, as no-one can get on to transfer their files.

In several such cases, it transpired that whenever they turned their machine on, the anti-virus software locked their machine as it had detected a virus. Since the users did not know what was happening, they talked to their departmental 'expert', who advised them to boot with a floppy disk. This circumvented the normal booting procedure of the machine and did not load the anti-virus software.

The employees continued to work without network or email facilities, and passed data around on floppy disk. Naturally, all of the machines were infected by the time the administrator found out what the problem was, and a significant amount of time and energy was needed to resolve the problem.

The upside of this is that the virus outbreak was contained to within a department. If floppy disks had been exchanged outside that department, however, this may no longer have been the case. If the anti-virus package had been configured to inform the users why the machine was stopping working when it did, the administrator would have been able to take action much earlier.

> *"there should also be another method of routine scanning whereby viruses are detected proactively"*

It is also recommended that error messages generated by anti-virus software are personalized to the company concerned. Most anti-virus software allows this, at least enabling additional information, such as 'call the Help Desk on extension 4431', to be displayed in any error dialogue boxes. If this is the case, the user will know what to do next.

Once again: do not get in the users' way unless you have to, and when you *do* have to, tell them what to do next, clearly and simply. Alerting the administrator directly via internal email or messaging is very advantageous, as the administrator will know when anything out of the ordinary, although not necessarily fatal, has occurred.

Pick a Package, any Package

When considering what type of anti-virus scanner to put in place, it must always be remembered that, when dealing on a corporate scale, one false positive for a given package could lead to a call to query it from every user in the corporation. Therefore, it is just as important that the package chosen generates a minimum of false positives as it is to detect the viruses required and to have a good update program.

Implementing any of these systems is naturally going to cost the company, both in financial terms and as regards time and effort in installation and maintenance. This needs to be considered against the question 'What happens if we don't do it?'

First, there *is* no correct answer to this. We all have a feeling that implementing virus protection software is going to help us in the long run: in general this is correct, in that we will all come across a virus outbreak at some point. However, if we are taking considerable pains to ensure our users' activity is not disturbed, we should at least think about what we would do if we implement nothing, thereby not disturbing the user at all, and a virus outbreak occurs: how then would we recover from the situation?

Recovery and Consequences

At a detailed level, the cost of recovering from a virus outbreak will rather depend on the virus concerned and the number of machines affected. If the outbreak is restricted to only a few machines then the cost of clearing up will be small. It will nevertheless involve time and effort in backing up data, reformatting hard disks, re-installing software from supplied floppy disks, and replacing the data. The more machines are affected, the greater the cost.

The question is, however, how will you know when you have a virus infection? Without some degree of virus protection available within the organization, it could go unnoticed for some considerable time, either until a machine starts to malfunction or (and in my opinion worse) a customer tells you that a floppy disk you sent them has infected their machines.

If a customer, as the result of such an incident, loses confidence in your supplies to him, it could mean a loss of business. This would have much larger financial consequences than the potential cost of early virus detection on your system. How do you know that it is you who has infected those client machines? You could spend time trying to track down a virus which does not even exist on your machines, and you might still lose the client.

A policy of virus detection and protection which is known and understood within a company will not only help you to trap a virus earlier, and therefore not send out an infected floppy disk to a client, but it will also enable you to have some authority to say to the client 'it cannot be us' when they report a virus to you – you will be able to point to the virus detection systems you have.

Costing the Clean-up

Purely in clean-up terms, and assuming an infected machine uses, maybe, four applications, the cost per machine to clean up after a virus will probably approach £250. This assumes all the software is readily available for a technical person to re-install on the machine, and that a tape streamer is available for backing up all the data from the machine.

The PC needs to be backed up (probably twice), the hard disk reformatted, the applications re-installed from floppy disk (which should include the operating system) and the data re-installed from the back-up tape. All being well, a good technician can probably do this in one day.

So, to fix one machine will take £250 worth of a technician's time. Add to this the loss of productivity of the person who normally uses that machine, which could be up to another £250. This does not take into account the fact that, if that person happens to be one who usually deals with clients, there could be potential problems filling client requirements, which may lose you the client. Further, invoices and payments may go out late, which may have additional consequences.

Multiplying this by a typical fifteen-person workgroup, and assuming an IT department of five working on the systems, it will cost nearly £4000 for the IT department, £4000 for the fifteen people in the department who cannot use their computers for one to three days, plus the opportunity cost to your organization of having this workgroup idle for that length of time. This could run to between 10 and 100 times the cost of the £12,000 (approximately) in core business costs for having the virus outbreak.

At that point, it becomes easy to justify spending considerable time and effort on the right hardware and software combination to help your organization remain in control of the virus problem.

How Much is Enough?

An added degree of complexity comes into the equation when we consider that, to be adequately protected from a potential virus attack, we should not put all our trust in one software product. This is because, in the ever-changing complexion of the virus world, we can best protect ourselves by having more than one source of virus information and protection.

On more than one occasion, I have seen a virus outbreak in an organization with a virus strategy in place, simply because the software on which the organization relied, across all machines and servers, came from a single vendor. This particular software had an engine which could not detect one specific virus type well; therefore, when this virus was introduced into the organization it was able to spread easily before it was caught.

This begs the question that, if we assume that more than one product should be available as the solution to the problem, how do we reconcile this with maintaining an environment which is easy for users to use and at the same time easy to maintain for the administrators?

The first step along this route is to provide IT staff who are likely to be visiting machines with a different virus protection tool from the one already installed on the machines being visited. We can assume that, if a virus is on a machine where virus protection is installed, the same piece of software on a floppy disk will probably fail to detect it the second time. Having a separate tool available to IT staff will reduce the chance of this happening. This will certainly highlight a new virus when problems are reported to IT staff, and will give the company a greater degree of confidence that viruses will be detected.

Alternative Routes

There should also be another method of routine scanning whereby viruses are detected proactively rather than reactively, with more than one piece of software. The best way to do this is to have scanning of the file servers done by software different from that in use on the desktop.

In order to maintain virus signature information in an up-to-date way, we will probably have to install the same scanner engine on the server and the clients to gain the benefit of downloading new signatures automatically.

The solution to this paradox is to have more than one scanner available to the server, which is then triggered manually or semi-automatically to scan at longer intervals than the core product in use on both the client and the server.

One way of achieving this would be to have the core product scanning on a nightly basis, and scanning all files which are saved and opened from the disk drives, but to have a secondary product which is brought into play manually on a regular basis, when the primary product is disabled, for example over a weekend.

“any users who copy applications to the file server should have their machines thoroughly tested on a regular basis”

Thus, the primary product will be in use day to day, and will maintain virus signatures at the client workstations, but during Friday evening the primary product is disabled and a secondary product on the server is enabled, which progressively scans the disk drives and alerts the administrators to any anomalies which it uncovers.

If a good secondary product is chosen and this is also installed on all servers across a network, the additional administration is small. The overhead on the file server is also small, since it is disabled for most of the time.

Virus detection using the normal tools available will scan local hard disks and floppy disks for suspect files, and server versions of the product will also scan server disks. However, most off-the-shelf packages in their native form will not scan files transferred between users over email (though naturally they will scan files as soon as they are saved to the hard disk).

Many packages are now addressing this issue, and providing engines running on the post office machines which inspect all email attachments, and scan them using the standard virus protection engine to determine whether they are virus free. There are also several stand-alone products which use a third-party virus-scanning engine to achieve the same aim; specifically, to protect an Internet connection.

These third-party products are a good system to consider. You can choose your virus-detection strategy based on your other requirements as an organization along with the ability of the virus detector to detect viruses, then leave the handling of email attachments to a third-party product which uses your chosen detection engine and which fits with your email strategy.

Make it Tough!

There are other measures that could and should be taken to ensure that viruses do not have an easy life within a corporation. These should be in place in any large company, with full-time network support, but it is worth reiterating a couple of simple measures which will help.

First, ensure the security on your network server is as tight as possible around all areas which contain shared applications. If users run applications off the server, and one user with the ability to write to the server has a virus which infects those applications, that infection can soon be prevalent on the executable file and therefore infect all machines which run it.

Any users who copy applications to the file server should have their machines thoroughly tested on a regular basis, since an executable with a virus copied to the file server can cause rapid spread of infection around an organization. It is normal to find an area of the network shared amongst workgroups, and it is very easy for someone in that workgroup to copy a file into that area for other users of the workgroup to run, especially if that user is reasonably technical and has a machine at home. Maintain a ‘weather eye’ on all these areas of the network, to ensure they do not pass viruses around unwittingly.

The security on a file server can, of course, be circumvented by the supervisor account. The supervisor account will have (in general) total and full access to all parts of the network file server. Thus, a virus-infected machine being used by the supervisor (or somebody logged in as a supervisor equivalent) could easily infect a file server.

It is good practice for a supervisor to have more than one login, and for the second login only to have standard user-equivalent rights. Thus, the supervisor can work most of this time, maintaining security, and only when necessary to do supervisory functions, would he log in with the full supervisor logon.

Backups and Other Safeguards

Of course, one of the best protections against a virus outbreak, apart from anti-virus software and monitoring, is a good backup system. This should be in place whether or not a company is thinking of the virus problem *per se*, and also for any issue which may involve recovering from a potential disaster. A virus outbreak is one of these issues in just the same way as fire or flood, and, like these, should have a good backup system as its core.

The backup system should be automatic and should be regularly checked and tested, or (a) it will not get done, and (b) when it does get done the data may not be valid.

In general, viruses will infect application files rather than data (although some of the recent viruses prove this is not always the case); therefore, as well as a good backup, the administration department should have access to the applications in order to re-install them from floppy disk or CD onto users' workstations as necessary.

Having this available and, better still, having tested it, will make recovery a much cleaner and quicker operation. The faster you can recover from any disaster, the less exposed you will be as a company.

Another commonly-used technique is to have a 'dirty machine' which is off the network, and on which no company business is carried out, on which to try out new software or test floppy disks coming in from outside the company. Whilst a useful tool, especially in situations where software is coming in from the outside on a regular basis, it should not replace rigorous anti-virus procedures on client machines.

In instances where there are multiple machine types such as PCs, *Macintoshes*, and UNIX platforms, the issue becomes yet more complicated; however, the same principle still applies. More than one anti-virus package should be used wherever possible, and the method for maintaining virus signatures should be easy. Finally, the anti-virus solution should not get in the way of users until it is necessary.

Conclusions

In mixed environments, it will probably not be possible to resolve all these issues with a single package as a core product – some issues may have to be resolved outside the scope of anti-virus software. For example, it may be possible in a mixed PC and *Macintosh* environment to use the same scanning engine on both machines, and to have them both connected to servers such that the virus updates are sent to the different machine types automatically.

However, using commonly-available distribution tools, this issue can be addressed quite simply, and indeed may give increased flexibility throughout an organization which may wish for a different scanning engine on the server from that on the clients. This obviously addresses the problem of multiple virus scanners to minimize the potential of one scanner missing a virus.

Finally, it is imperative that you keep up to date with information available regarding the latest threats and virus prevention techniques. Any corporate virus strategy needs at least one subscription to *Virus Bulletin*!

Phil Crewe has been a member of *VB's* advisory board since its inception in 1989, and has written many articles in the past. He can be contacted by email at philcr@hoima.i-way.co.uk

PRODUCT REVIEW 1

InocuLAN for Windows NT

Martyn Perry

Cheyenne Software, well known for its *ARCserve* products, steps forward this month with *InocuLAN for Windows NT*. Under evaluation is the Live Trial version with signature updates, with a licence for a single server for thirty days. After this time, the files are automatically disabled.

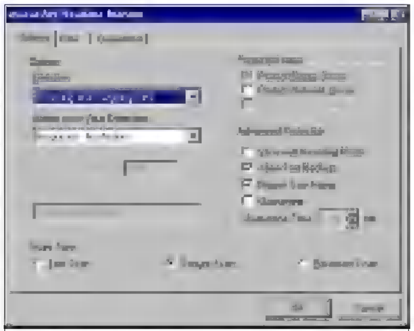
Presentation and Installation

The product is supplied on CD-ROM with a hard copy user guide. The installation autoloads SETUP.EXE when loaded in the CD-ROM drive. First, the licence information is displayed, then the option to install *Acrobat 2.1* is offered: This gives the facility to read some of the on-line documentation, which can be found in the directory ONLINE.DOC.

The installation platform is selected from *NetWare*, *Windows NT* or Client. Selecting *InocuLAN for Windows NT* displays the installation components, then a further set of products which can be installed onto *NT*. The evaluation version chosen was *InocuLAN v4.0 for Windows NT* Build 216 with signatures v3.24. The virus signatures were subsequently updated to version 3.34.

The installation now requests a licence key, following which a user name must be entered. If a previous version of the product is detected, the existing settings may be kept in place, or they may be overwritten with defaults specified in a configuration file (INOCULAN.ICF). There are further options to install Internet Plug-ins, Start-up options and *NetWare* Domain management support.

The final option decides which set of components to install; namely: Express Setup (installs *InocuLAN*, Autodownload, and Alert Software with associated program groups and icons), Custom Setup (installs *InocuLAN* with manual selections, allowing the various components to be selected individually: this option is necessary to install Domain Management), and Remote Setup (installs onto remote machines).



Real-time scanning can cope with various combinations of files.

Choosing the default Express option, the installer is next prompted to select the target directory for *InocuLAN* (C:\INOCULAN) and the Alert home directory (C:\Alert). Installation automatically detects any Internet components present, and decides whether these are to be integrated into the product. This is a useful feature, since *InocuLAN NT* is automatically configured to work with *Netscape Navigator* or *Microsoft Internet Explorer*.

At this point a choice can be made to add Real-time Quick Access monitor to the Start-up group. The program then installs the appropriate modules, creates the program groups and configures the required registry settings.

The Real-time device driver does not take effect until the computer is restarted. This option is presented so the user can restart immediately, or wait for a more suitable time. If an error occurs during installation, Dr. Watson for *Windows NT* creates an application error log.

Getting Started

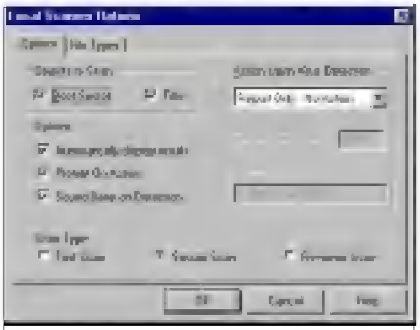
InocuLAN's operation is based on defining and running jobs. These jobs define the options available to perform virus scanning and handle the results of the scan. Three levels of virus scan are available: Secure, Fast, or Reviewer.

Secure Scan scans the complete file, whereas Fast Scan only checks the beginning and end of each file. The Reviewer Scan is used to detect inactive or modified viruses. This mode is more prone to generate false alarms, and should thus be used with care, and with reporting option only, to avoid damaging a clean file. Its prime purpose is to check the system if a virus is suspected, due to system behaviour, but none is detected with the current signature set.

The files and directories for scanning can be selected using the *Windows NT* interface. While the scan is proceeding, a running total is displayed showing the progress of the scan and the file currently being scanned. In addition, directories which have been scanned are shown with a tick against the directory name, while a directory undergoing a scan has a magnifying glass next to it. The scan can be started and stopped from the console. The scanner has three modes of scan operation: On-access, Immediate and Scheduled.

On-access Scanning

The on-access scan can be accessed from the system tray (right click on the icon in the system tray next to the clock) to display a menu, which allows immediate selection of the type of real-time scan monitoring (incoming only, outgoing only, incoming and outgoing together, completely disabled). The user can also choose the type of scan to perform (Fast, Secure or Reviewer), and the action to be taken on a file if a virus is detected (broadcast with no action, delete, copy and cure, rename, move, purge, rename, and move – see below for explanation).



On-access scanning offers many options from which to select.

Protected areas can be selected to cover the floppy drive for boot sector scanning and network drives to scan files moving between mapped drives. *MS Exchange*, if installed, can also be protected. Additional facilities can also be selected.

The option 'Virus Wall Incoming' stops incoming infected files being copied from the workstation to the server. 'Allow Fast Backup' permits backup software to archive files without additional real-time scanning. 'Report user name' allows the administrator to know which user has been trying to copy an infected file.

'Quarantine' ensures that if a user attempts to copy or execute an infected file, he can be blocked from further access to the server for a defined time. The administrator can grant users access again by removing the user's name from the Quarantine screen.

All files, or files with specified extensions, may be scanned. The default set of extensions are the same as the immediate scan, with the exception of BIN. Compressed files with ARJ and ZIP extensions can also be scanned. The final option excludes specified file extensions.

Immediate Scan

The default executable file extensions are APP, BIN, COM, DLL, DOC, DOT, DRV, EXE, OVL, OVR, PRG, SYS, VXD. Actions available are:

- Broadcast – no action
- Delete – automatically removes the file
- Copy and Cure – copies infected file to the directory INOCULAN\VIRUS before attempting to clean the file. If unsuccessful, the file is automatically renamed with the extension AVB
- Rename file – changes the file extension to AVB. If there is more than one file with the same name, it uses the extension AV#, where # is 0, 1, etc
- Move file – moves file from its current directory to the quarantine directory INOCULAN\VIRUS
- Purge file – deletes a file irrevocably
- Rename and Move file – changes the file extension to AVB and moves the file to INOCULAN\VIRUS

Scheduled Scan

This option can select target directories and sub-directories for scanning – a list of exclusions can be created. To ease the burden on the CPU during a scan, CPU usage level can be set depending on performance required.

The scheduled scan can be run at start-up or at pre-defined intervals (monthly, daily, hourly with a selectable start time). If the repeat time is set to all zeros, the scheduled scan is run only once.

Administration and Domain Management

If logged in with Administrator rights, no additional password is required to access the scanner administration. The administration is divided into four sections: Domain Manager (Domain and Server Automated Scanning), Domain Manager for *NetWare* (same as above but for *NetWare*), Local Scanner (Immediate Local and Mapped Drive scanning), and Service Manager (Start, Stop and Configure InocuLAN services).

The Domain Manager has several functions: create, view, modify and delete a domain; create a point-to-point connection with a remote computer; view the summary information for the domain or an individual server; and view the event log. Domain Manager for *NetWare* provides the same domain management facilities under a *NetWare* environment. The local scanner can change the scanner options, view the scanning log, and check the version information.

The Service Manager provides support to configure the *InocuLAN* services. This includes defining how some of the basic services are run:

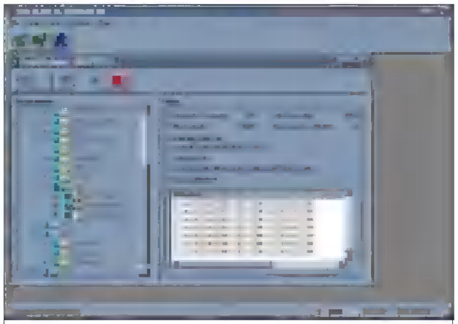
- Selecting the background services to be started automatically on boot-up or under manual control
- Define how many days a completed job should remain in the Job Queue, giving a supervisor the chance to see, for example, the jobs run over the previous week
- Active Server time out can be used to define the number of minutes *InocuLAN* should wait before considering a server inactive, because it failed to receive a 'heart-beat' signal (see below) from the server

The event log can be configured to set the number of messages to hold in the log at any one time, to set how many days a message should stay in the log before being automatically purged, and the type of message to store. Message types are: critical (default; warns of a virus or problem with the service), warning (warns if a file is skipped; reports other non-critical information), and informational (logs that the service has started or stopped and if no viruses were found). The Scan Log has a similar set of options to the Event Log.

Configure the Service Broadcasts

InocuLAN provides the facility to allow servers to 'advertise' themselves on the network. This can be done using Mailslots protocol, TCP/IP protocol or a combination of both. The information that can be broadcast includes status changes, signature versions, engine versions, real-time and scheduled status changes and the OS version on the server. This allows a supervisor to keep track of the anti-virus protection offered on the various servers on the network.

All this ability can be achieved with no administrative intervention, using the network Auto Discover. This can be used to find servers on the network and store their details in the *NT* Domain table, the IP mask table, and the IP subnet table. DOMAIN.TBL, the *NT* Domain table, stores new *NT* domains and allows broadcasts via Mailslots to all machines in the listed Domain. IPMASK.TBL keeps the local IP mask. The IP Subnet table (IPNET.TBL) keeps track of the IP subnets, which allows *InocuLAN* to make broadcasts via IP.



Virus detection was high, with the polymorphic rates being lowest, at 94.7%.

Auto Discover updates the above tables at intervals that can be configured by the network administrator. The 'heart-beat' broadcast from each server is used to check that it is still active on the network. The amount of broadcast information indicates additional network traffic. If this is a problem, Auto Discover can be disabled and the information loaded manually into the tables. Also, to help reduce network traffic, the interval between 'heart-beat' signals can be configured.

Finally, it is possible to synchronise the broadcast configuration for all servers on the network from the settings of one InocuLAN server by editing the Registry.

Reports, Activity Logs, and Updates

If the Alert Manager was included at installation time, when a virus is detected, alert messages can be sent via Network Broadcast, *MS Mail*, *MS Exchange*, SNMP, Trouble Ticket to a specified printer, or Pager depending what has been set up in Alert. The message will also be shown in the Scanning Log and the *Windows NT* Event Log.

To provide update support, the files AVH32DLL.DLL, VIRSIG.DAT, FILELIST.TXT, VIRUS.LST, and VIRINFO.DAT must be updated. *InocuLAN* has three levels of automated updating. First, updates are downloaded from the *InocuLAN* update site via FTP or modem connection to the designated update server. They are then automatically downloaded to other servers on the network using preset parameters controlling the update process. Finally, the workstations obtain updates from the servers when they log in to the Domain provided the program AVUPDATE is run as part of the login script.

Detection Rates

The scanner was checked using the test sets: In the Wild, Standard, Polymorphic and Boot Sector (see summary for detail). The tests were conducted using the default scanner file extensions supplied, and infected files were set to delete. The residual file count was used to determine the detection rate. Overall, detection was excellent.

Only one Boot Sector sample (Ornate) was missed. Six samples went undetected from the Standard test-set: one of Greets.3000, three of Maresme.1062, and two of Positron.

Initially, In the Wild only missed four samples of Laroux, since XLS was not a default file extension when first run, but when the scan selection was changed to all files, these were detected correctly. The main problem was in the Polymorphic samples: although only a few samples of DSCE.Demo, Neuroquila and Russel.3072 were missed, about 20% of Sepeltura samples were missed, as were all samples of Girafe:TPE.

There appeared to be one false positive using the secure mode. HLLP.4075 was reported in UPACKEXE.EXE file. This was from a clean system and was not detected by several other scanners which were tried.

Real-time Scanning Overhead

To determine the impact of the scanner on the workstation when it is running, we timed how long it took to copy 200 files of 21.24MB (EXE and COM files) from one directory to another using XCOPY. The directories used for source and target were excluded from the scan to prevent a file being scanned while waiting to be copied. The default setting (Maximum Boost for Foreground Application) was used for consistency in all cases. Due to the different processes which occur within the server, the tests were run ten times for each setting and an average taken. The tests were:

- Program not loaded – establishes the baseline time for copying the files on the server
- Program unloaded – run after the other tests to check how well the server is returned to its former state
- Program loaded without Incoming/Outgoing on-access tests running – tests the impact of the application in a quiescent state
- Program loaded with just Incoming on-access checks running using Fast Scan – tests impact of the real-time scan for just reading the files
- Program loaded with just Incoming on-access checks running using Secure Scan – compares the effect between the two scan modes
- Program loaded with Incoming/Outgoing on-access checks running and Secure Scan – shows the full overhead of the real-time scans
- Program loaded with Incoming and Outgoing on-access checks; Secure scan in operation; Immediate scan running – full impact of running real-time and immediate scanners on files. See table for detailed results.

Summary

InocuLAN for NT provides a comprehensive selection of installation options. Nevertheless, installation is easy to perform. The range of scanner options is extensive and the virus detection rates very good. In addition, there is a range of facilities for server-to-server communication. This is not limited to broadcasting alert messages, but encompasses the ability for an administrator to monitor the status of individual servers as well as provide automated scanner updates across a security domain.

This is a product which combines the performance of a good scanner with the facility for an administrator to manage a complex network of servers, including remote installation and enterprise roll-out, from a single point – a commendable achievement.

InocuLAN for Windows NT

Detection Results

Test-set[1]	Viruses Detected	Score
In the Wild File	509/509	100.0%
In the Wild Boot	87/88	98.9%
Standard	759/765	99.2%
Polymorphic	11366/12000	94.7%

Overhead of On-access Scanning:

The tests show the time (in seconds) taken to copy 200 EXE and COM files (20.55MB). Each test is performed ten times, and an average is taken.

	Time	Overhead
Program not loaded	13.4	–
Program unloaded	14.7	9.8%
Program loaded		
No Incoming/outgoing files, no manual scan	19.1	43.1%
Incoming files (fast), no Outgoing files, manual scan	24.2	80.9%
Incoming files (secure), no outgoing files, no manual scan	24.2	81.1%
Incoming and outgoing files (secure), no manual scan	24.7	84.7%
Incoming/outgoing files (secure), manual scan (secure)	29.1	117.8%

Technical Details

Product: *InocuLAN for Windows NT v4.0.*

Developer/Vendor: *Cheyenne Software*, 3 Expressway Plaza, Roslyn Heights, NY 11577 USA. Tel +1 516 465 5700, fax +1 516 465 5313.

Distributor UK: *Cheyenne Software*, Furness House, 53 Brighton Road, Redhill, Surrey RH1 6PZ. Tel +44 1737 775500, fax +44 1737 775520.

Price: In the UK, £695, including a 25-user client pack. Supports *Windows NT 3.5x* or *4.0.* The client pack can be expanded to 100 or 250 users, and installation is permitted once on a server while the client packs are not affected. Server or workstation company-wide administration possible. Updates to the package are available through the company Bulletin Board; new patterns being posted monthly.

Hardware Used: *Compaq Prolinea 590*; 80MB RAM with a 2GB hard disk, under *Windows NT v4.0* with service pack 1.

[1]**Test-sets:** For a complete listing of all the viruses used, see *VB*, May 1997, p.20.

PRODUCT REVIEW 2

McAfee VirusScan

Dr Keith Jackson

McAfee. A name synonymous with anti-virus products since the very early days. I have reviewed this company's software for *VB* twice before (1993 and 1995) – over the years, the software has changed beyond all recognition.

Documentation

The physical documentation provided with the review copy was a 32-page booklet which concentrated on providing installation instructions – these vary slightly for each of the operating systems on which *VirusScan* is available.

The details provided to help with installation are simple: most of the installation information is provided in the form of onscreen information, and not described in the booklet. Also included in the booklet are 10 pages which describe briefly how to prevent virus infection, what to do if a virus is found, how to carry out a scan, and how to update the scanner. The booklet's authors have been very thorough, providing voluminous *McAfee* contact information.

McAfee claims that *VirusScan* is a 'powerful and advanced desktop anti-virus solution'. Notwithstanding the adjectives advertising people feel compelled to inject, I am surprised by the appearance of the word 'desktop' in that definitive quote. Does it mean it is not meant to work on laptop PCs?

The on-line help provided is well written, and describes each option succinctly. The somewhat terse style may not be to every taste, but I liked it. I get fed up of wading through acres of marketing spiel to get to solid information.

Installation

VirusScan was provided for review on CD-ROM, and on six 1.44MB 3.5-inch floppy disks. The CD contained versions of *VirusScan* for *Windows 3.1*, *Windows 3.11*, *Windows 95*, *Windows NT*, *OS/2*, and DOS. I opted for the CD-ROM format, and first installed the *Windows 95* version.

Installation proved easy – the only tricky bit was ensuring that the selection of subdirectory on the *McAfee* CD-ROM corresponded to the operating system in use at the time. *McAfee* agreed with this point, and version 3.0.1 contains a CD Autorun, which should make things easier.

Once SETUP.EXE was executed from the appropriate subdirectory, installation involved merely following onscreen instructions, some of which were deceptively simple. Perhaps too simple. For instance, a choice was required between 'Typical – Recommended', 'Compact – Minimum Required Options', and 'Custom'. But what *are* the available options? We are not told, making the choice a bit hit and miss. *VirusScan* installation threw up a message box showing a 'System File Error' with five DLL files, and provided a warning for two other files. I'm not sure what caused this; however, ignoring it didn't seem to cause anything drastic.

Beavering on, I selected the 'Recommended' setup, and was informed of the changes to be made. Installation then started transferring files, and maintained a pretty, but ultimately quite useless, set of three bar graphs in the bottom right-hand corner of the screen to show how far things had progressed.

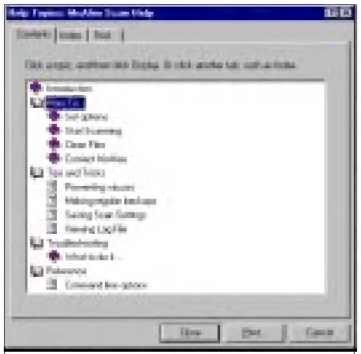
A scan of my PC was carried out, and the installation program then requested that a blank floppy disk was inserted so that it could create an 'Emergency Disk'. The latest information about *VirusScan* is then shown onscreen, an event that informs the user that *VirusScan* now incorporates 'Hunter scanning technology'. The computer must be rebooted before *VirusScan* changes take effect.

Apart from help files, only two of the installed icons referred to executable programs. One executed the main scanner; the other allowed memory-resident software to be configured.

At a later stage, I reinstalled the *Windows 3.1* version of *VirusScan* on my test PC. The two installation programs were very similar, though the main scanner provided with this version looks different from the *Windows 95* version: disk 'profiles' are available, single button scanning for hard disk and floppy disks is available, a scheduler is available, and the menu selections differ greatly. The newest *Windows 3.1* version, according to *McAfee*, has a new GUI which is exactly the same as that for the *Windows 95* version.

When this version of *VirusScan* was installed, three executable programs were available – the two described above for the *Windows 95* version, and one which provided information on the current status, and statistics of files scanned.

The DOS version of the product was much quicker to install than either *Windows* version. Fewer files were copied, and the changes to AUTOEXEC were offered, but not enforced. During DOS installation, the hard disk of my test PC was scanned, and some viruses (which had been left behind from the previous *Windows 3.1* testing) were found. Next, the installation program insisted on having a clean disk inserted into the floppy disk drive, presumably to enforce a known clean system. This was a nice touch. All in all, I had no real problems



The on-line help came in for some high commendation.

with any of the installation programs, and the software did appear to be doing its best to be helpful at all times. Very good.

Speed

Using the default settings, the *Windows 95* version scanned drive C of my test PC in 24.9 seconds. Only program files were inspected, and 419 files (out of 1406) were examined. Scan time increased to 54.9 seconds when all files were scanned, and further to 60.6 seconds when checks were also made for compressed files. The 'compressed files' option increased the scan time even though there were no compressed files present on the test PC.

Curiously, apart from a lack of activity, the only thing to indicate that *VirusScan* had finished was an information bar at the bottom of the dialog box stating 'No viruses found'. No splash screen, no large banner, no request to push a button to acknowledge the results. This stood out from the rest of this version, where much attention seemed to have been paid to its appearance.

The *Windows 3.1* version of *VirusScan* scanned the hard disk of my test PC in 37.4 seconds, but as the hard disk had been reconfigured for *Windows 3.1*, this scan only inspected 395 files (out of 655). A Turbo button was available, but this did not seem to make any difference to measured scan time. Digging around in the help files explained this – the Turbo button only selects a certain combination of scan options (executables only, and scan on all subdirectories). The DOS version took 40.2 seconds to scan the hard disk; roughly the same speed as the *Windows 3.1* version.

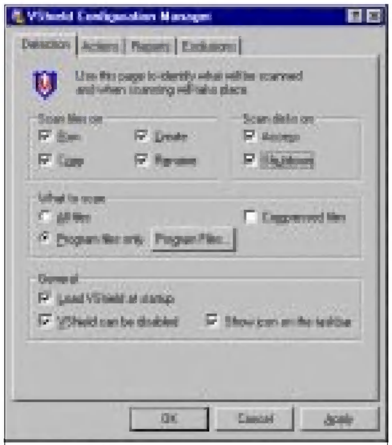
Normally, I include the scanning speeds of other scanners for comparison purposes. However, *VirusScan* is available in several different guises, and so many different comparative figures would have to be quoted, that any underlying meaning would be obscured. Therefore, for this month (and for all multi-platform scanners) I have abandoned my usual practice of including comparative scan times.

Scanning

Using *VirusScan* is straightforward: the executable program contains the usual *Windows* drop-down menus, and buttons for quick access to oft-used features. The scanner can be tailored to inspect program files only (defined on a tailorable list of file extensions), all files, and compressed files.

The *Windows 95* version of *VirusScan* detected all 476 of the In the Wild samples, and 530 of the 532 samples of the Standard test-set (missing Power_Pump.1, and one sample of Cosenza.3205). Both are good results, very close to 100%. All boot sector viruses were also detected.

A total of 12,098 samples in the *VB* test-set were scanned. The scan report indicated that 12,083 of these were infected, leaving fifteen samples not found infected. Subtracting the two missed in the Standard set, *VirusScan for Windows 95* claimed to find all but 13 of the 11,000 polymorphic samples were infected (99.9%). Impressive.



VShield Configuration Manager

Detection | Actions | Reports | Exclusions

Use this page to identify what will be scanned and when scanning will take place

Scan files on
☑ Run ☑ Create
☑ Copy ☑ Rename

Scan disks on
☑ Access
☑ Shutdown

What to scan
○ All files
● Program files only [Program Files...]
☐ Compressed files

General
☑ Load VShield at startup
☑ VShield can be disabled ☑ Show icon on the taskbar

OK Cancel Apply

VSHIELD is a useful and efficient addition to *VirusScan*

Is the *Windows 3.1* scanner as good? The first thing that I noticed when using this version of *VirusScan* was that a warning message 'The file infection list is full' was produced after approximately 1600 infected files had been found. The infection list had to be manually cleared (by pressing a button) before scanning could continue. As over 10,000 files in the *VB* test-set were found to be infected, I had to do this many times before the scan could reach its conclusion. This will, however, rarely affect 'average' users.

The *Windows 3.1* version of *VirusScan* found 10,223 of the 12,098 samples infected. It reported that 461 out of 476 viruses from the In the Wild set were detected (96.8%). Similarly, Standard detection was 475 out of 532 (89.3%), and polymorphic detection was 9197 from 11,000 (83.6%). This version also detected all boot sector viruses.

The DOS version detected 11,438 of the 12,098 test samples as infected. Different from both other versions above. I have no idea why this is so – it needs evening out.

False Positives and Reports

I tested *VirusScan* against the *VB* false positive test-set. This comprises 5500 executable files, held on CD-ROM, which have been culled from well-known software products. The *Windows 95* version, as well as the *Windows 3.1* and the DOS versions, checked the entire disk and, correctly, did not find a single file that it deemed to be infected by a virus.

The *Windows 3.1* version seems to create a report file on disk only if 'Scan' is selected, not if a 'Profile' is selected. Is this sensible? It took me ten minutes to figure out why I could never find a report file on the hard disk of my test PC. I was innocently using 'Profiles', which scanned the hard disk but (unbeknown to me) did not do the same as using the 'Scan' button. 'Profiles' were conspicuous by their absence from the *Windows 95* version.

Memory-Resident Software

VirusScan's memory-resident software, VSHIELD, can be configured in many ways to detect viruses during execution, creation, copying or renaming of files. These options can be easily configured using a standard *Windows* utility, and take immediate effect. By default, the *Windows 95* version excludes the 'Recycle Bin' from VSHIELD's gaze.

So how good is VSHIELD at detecting viruses? I copied the entire test-set from CD-ROM to hard disk, instructed VSHIELD to move infected files to a folder, and waited to see which of the files being copied were detected as infected.

For the *Windows 95* version, the answer was encouraging. All but two In the Wild samples were detected (both samples of One_Half were missed), and all but three of the Standard samples were detected (Power_Pump, and one each of Cosenza.3205 and Dei.1780 were missed). Note how close these results are to the results of the main scanner. All bar 656 of the 11,000 polymorphic samples were found; a detection rate of 94%, including failing to detect the 500 samples of PeaceKeeper.B.

The *Windows 3.1* version of VSHIELD was not as good: it missed 15 In the Wild samples (97%), 45 Standard samples (91%), and 1799 polymorphic samples (84%). The polymorphic rate is biased by a failure to detect any samples of DSCE.Demo or PeaceKeeper.B. Although acceptable for a memory-resident scanner, these results are not as good as the *Windows 95* version.

Curiously, the *Windows 3.1* version of VSHIELD detected all 500 samples of One_Half, but the *Windows 95* version failed to detect two samples of this virus.

Overhead

VSHIELD must impose an overhead on program execution: if it doesn't then it is probably not doing much! I tested this by copying the entire In the Wild test-set to hard disk.

Without VSHIELD present, copying the 476 files took 26.3 seconds. When the *Windows 95* version of VSHIELD was set to scan only when a file was renamed (i.e. it was looking at the files but finding nothing to scan), the test time rose to 33.0 seconds, and still 476 files were copied. When all VSHIELD scan options were activated, the test time was 33.0 seconds, but only two files were actually copied.

Although these figures show that VSHIELD can introduce a severe overhead, the results are encouraging. VSHIELD can be tailored to find a suitable balance between performance and overhead, the detection rate (see above) is very good, and even when all checks are active, I have reviewed products that impose a far more onerous overhead than VSHIELD does. All in all, well done to the *McAfee* developers.

Conclusions

VirusScan is easy to use, comes with versions for most OSs, detects viruses well, and performs at a reasonable speed. The memory-resident software provided is very good at detecting viruses, and very configurable. I doubt anyone would be embarrassed by purchasing *VirusScan*.

Having said that, I do despair at the effort that has gone in to producing a fancy front-end for what remains after all merely a utility; something that detects viruses. I am not singling out *McAfee* in this respect; many (perhaps even most) anti-virus programs seem to be heading this way. The question must, however, be asked: is this really what the users want?

Be aware that the various scanner versions have quite different detection abilities. The *Windows 3.1* version was worst, maybe because (for some odd reason) it was version 2.5.3 – everything else had been upgraded to version 3. Even things having seemingly the same version (the *Windows 95* scanner and the DOS scanner) have disparate detection rates. I haven't a clue why this should be, but it is indisputable. If you're considering purchasing the product, ask questions about this.

That being said, *McAfee* has stated that all issues concerning the *Windows 3.1* version have now been addressed: readers should apply to *McAfee* for information on the latest release.

Technical Details

Product: *McAfee VirusScan v3.0* (*Windows 3.1 v2.5.3*).

Serial Number: EOE3-4UYU-2M6Z.

Developer/Vendor: *McAfee Inc*, 2710 Walsh Avenue, Santa Clara, CA 95051-0963, USA. Tel +1 408 988 3832, fax +1 408 970 9727, BBS +1 408 988 4004.

Availability: DOS, *Windows 3.x*, *Windows 95*, *Windows NT*, and OS/2. System requirements vary, but at least 2.5MB of RAM is needed.

Price: RRP for a single-user licence is US$49.00. This includes dual media (CD and floppy) and support for all five platforms, free DAT updates for the life of the product, and electronic upgrades (full product releases) for one year. Subscription pricing also available – apply to *McAfee* for details.

Hardware used: A 133 MHz Pentium with 16MB of RAM, a 3.5-inch floppy disk drive, a CD-ROM drive, and a 1.2GB hard disk divided into drive C (315MB), and drive D (965MB). The PC can be configured to operate under *Windows 95*, *Windows 3.11*, *Windows 3.1*, or DOS 6.22.

Viruses used for testing purposes: Where more than one virus variant is available, the number of examples of each is given in brackets after the virus name (if greater than one). A complete explanation of each virus, and nomenclature used, can be found in the lists of PC viruses published in *Virus Bulletin*. Details of the Standard, In the Wild, and Polymorphic sets are in *VB*, March 1997, p.17.

The boot sector test-set contains one each of the following 90 boot sector viruses: 15_Years, AntiCMOS.A, AntiCMOS.B, AntiEXE, Boot.437, BootEXE.451, Brasil, Bye, Chance.B, Chinese_Fish, Crazy_Boot, Cruel, Da_Boys, Defo, DelCMOS.B, Den_Zuko.2.A, Diablo_Boot, Disk_Killer, Empire.Int_10.B, Empire.Monkey.A, Empire.Monkey.B, EXEBug.A, EXEBug.C, EXEBug.Hooker, FAT Avenger, Finnish_Sprayer, Flame, Form.A, Form.C, Form.D, Frankenstein, Galicia, Hare.7750, Ibex, Int40, J&M, Joshi.A, Jumper.A, Jumper.B, Junkie, Kampana.A, Leandro, Michelangelo.A, Moloch, Mongolian_Boot, Music_Bug, Natas.4744, Neuroquila, NYB, Ornate, Paula, Parity_Boot.A, Parity_Boot.B, Pasta, Peter, QRry, Quandary, Quiver, Quox.A, Ripper, RP, Russian_Flag, Sampo, Satria.A, She_Has, Stealth_Boot.B, Stealth_Boot.C, Stoned.16.A, Stoned.Angelina, Stoned.Azusa.A, Stoned.Bravo, Stoned.Bunny, Stoned.Daniela, Stoned.Dinamo, Stoned.June_4th.A, Stoned.Kiev, Stoned.LZR, Stoned.Manitoba, Stoned.No_Int.A, Stoned.NOP, Stoned.Spirit, Stoned.Standard, Stoned.Swedish_Disaster, Stoned.W-Boot.A, Swiss_Boot, Unashamed, Urkel, V-Sign, WelcomB, WXYC.A.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery:

UK £195, Europe £225, International £245 (US$395)

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, The Pentagon, Abingdon Science Park,
Abingdon, Oxfordshire, OX14 3YP, England

Tel 01235 555139, International Tel +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email: editorial@virusbtn.com
World Wide Web: http://www.virusbtn.com/

US subscriptions only:

June Jordan, *Virus Bulletin*, 590 Danbury Road, Ridgefield,
CT 06877, USA

Tel +1 203 431 8720, fax +1 203 431 8165

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

END NOTES AND NEWS

The **24th Annual Computer Security Conference and Exhibition** will be held in Washington DC from 17–19 November 1997. The event will feature over 120 sessions covering such topics as Network Security, Encryption, and Product Issues. Information can be found on the CSI's Web site; http://www.gocsi.com/.

The MIS Training Institute is sponsoring a conference on ***Audit and Security of Intranets***, from 18–20 August 1997, in Surrey, England. Amongst many others, such topics as intranet management challenges, viruses and Trojan horses, and firewalls will be addressed. For further details, contact Patricia Fischer on Tel +44 171 779 8292, fax +44 171 779 8293.

The ***Secure Computing* Awards** Ceremony, held on 29 April 1997 in London, judged *McAfee VirusScan* to be the best anti-virus product. The package also took the award for best security software. Best Backup was awarded to *Cheyenne's ARCServe*, and *Symantec's Norton Utilities* won the Best General Security Product category. For more detailed information on the awards, contact *Secure Computing* on Tel +44 1792 324000, fax +44 1792 324001.

***CompSec 97* will be held on London** from 5–7 November 1997. The conference is aimed at helping to highlight the risk to IT systems, assess security shortcomings, and protect against fraud, disaster, and negligence. Information is available from Amy Richardson at *Elsevier Science*; Tel +44 1865 843643, fax +44 1865 843958, or email a.richardson@elsevier.co.uk.

***Sophos Plc's* next round of anti-virus workshops** will be on 9/10 July 1997 at the training suite in Abingdon, UK. The company's training team is also hosting a Practical *NetWare* Security course on 3 July 1997 (cost £325 + VAT). Another initiative sees the company throwing open its doors to any organization wishing to evaluate anti-virus software. The move is aimed at helping administrators of multi-server networks to see how they can best implement virus protection within their organization. Information is available from Julia Edwards, Tel +44 1235 544028, fax +44 1235 559935, or access the company's World Wide Web page; http://www.sophos.com/.

Symantec is to create a '**global network of virus research centres**', known as SARCs, set up to identify and collect local viruses and contribute to product development. Based at the Santa Monica headquarters, SARCs are already open in the Netherlands and in Tokyo, and will open soon in Australia. The company has also announced the launch of the latest version of *CrashGuard 2.0*, designed to combat the loss of data due to application crashes. For information on these and other initiatives by the company, visit the *symantec* Web site at http://www.symantec.com/.

Dr Solomon's Software Ltd (formerly *S&S International*) is presenting **Live Virus Workshops** in the UK on 10/11 June, and 15/16 July 1997. The company has also been judged the best UK IT Company of the Year at a recent competition, beating fellow finalists *Logitech* and *Softimage Ltd*. Details from Melanie Swaffield at *Dr Solomon's*; Tel +44 1296 318700, Web site http://www.drsolomon.com/.

TouchStone UK announces the release of the **newest version of the anti-virus software *PC-cillin***. Developed in conjunction with *Trend Micro Inc*, the package, known as *PC-cillin DeLuxe*, includes technology to cover the risk of infection through the Internet. Further, this release is said to feature 'exclusive patent-pending technology to detect and clean all types of macro viruses'. The package includes an ActiveX Web browser with a personal link to *PC-cillin's* Internet Virus Lab. For further information, contact *TouchStone* in the UK; Tel +44 181 875 4456, or email Jackie Vause (vausey@flapjack.com) of *Flapjack Communications*.